

# Open Bike Initiative Setup Guide

This document was created by Hafner Industries and WTA to guide anyone who wants to set up a bike share platform similar to the one used in the Westside Bike Share Pilot Program.

This simple, open source model is based on technology designed by the Open Bike Initiative (OBI), an ad hoc project led by a group of Intel employees in 2013.

OBI technology can be used with any type or number of bikes. The system relies on using standard combination locks (we recommend combination u-locks for security). To check out a bike, the user sends a text with the bike number (printed on the bike) to a designated phone number, and they receive a reply text with the combination to unlock the bike.

This guide provides the directions to set up all the technical elements needed to send and receive the automated texts, register users, and set up a repair reporting mechanism. With this guide, some bikes, and an equal number of combination locks you can start your own bike share system.

Note: the guide uses a fictional bike share program called “Twig” to take you through each step. An internet search for “Twig Bike Share” will bring up the code and related pages.

**WTA can advise on other aspects to consider when setting up a bike share. Please call or email to set up a time to discuss your plan:  
503.906.7961 / [wta@wta-tma.org](mailto:wta@wta-tma.org)**



# Twig Bike Share

Powered by the Open Bike Initiative

## Setup Guide

### I. Objective

- The objective of this document is to guide you through the process of setting up a functioning version of the Twig Bike Share platform.

### II. Requirements:

- There is no technical know-how requirements to setting up this platform, but there are quite a few detailed steps. This guide was developed to be reasonably detailed as well as giving a general overview of the technology used to provide you with a straight-forward setup process, but sometimes things change with some of the web platforms that are used. You may need to spend some time reading documentation to get it up and running.
- At this time the system is designed for use in the USA. It may work in other countries but has not been tested.
- You will need:
  1. An internet connection. This is a web platform!
  2. A major credit card to setup an account with some of the online services. Costs are designed to be low, but you will still need to pay for a few things.
  3. A registered domain name. If you don't have one you would like to use for your service, one can be purchased through any domain name service. I recommend: <https://porkbun.com/>
  4. You will probably want a text editor open on your computer to record different bits of information along the way.
- Approximate setup time: **2-3 hours**

## III. Guide

### 1. Purchase a domain name

- **Why:**

To use services like mailgun we need a registered to domain name to send email as. This will also serve as the url of our web portal. ***If you already have a domain name you would like to use you can skip this step*** (as long as you have the ability to edit the DNS zone file for the domain).

- **Steps:**

1. Create an account on Porkbun: <https://porkbun.com/account/login>
2. Purchase the domain name of your choice.

### 2. Setup database with Mlab

- **Why:**

Mlab is a cloud based database service, with an excellent free tier. We will use this to store data from our app.

- **Steps:**

1. Create an Mlab account: <https://mlab.com/signup/>
2. Create a new database
  - Select Cloud Provider: *Amazon Web Services*, Plan Type: *Sandbox*, Region: *US East* or *Europe* (depending on which is closer to your location).
3. Create Database User
  1. Click on your new database on your account home page under *MongoDB Deployments*.
  2. Click the *Users* tab
  3. Click the *Add database user* button
  4. Fill out form and save these credentials for later
    - Make sure your password is long (over 15 characters) and only contains letters (upper and lower case) and numbers. Other symbols can cause problems during later steps.

4. Copy the line below where it says: *To connect using a driver via the standard*



```
Home
Database: twigsetup

To connect using the mongo shell:
% mongo ds243345.mlab.com:43345/twigsetup -u <dbuser> -p <dbpassword>

To connect using a driver via the standard MongoDB URI (what's this?):
mongodb://<dbuser>;<dbpassword>@ds243345.mlab.com:43345/twigsetup
```

*MongoDB URI:*

5. Add in your data base username and password

- *Example:*

If your username // password is:

```
twigdbuser // aGDIVDlVn9E3hEu83vDUz
```

- The value for this example would be:

```
mongodb://twigdbuser:aGDIVDlVn9E3hEu83vDUz@ds243345.mlab.com:43345/twigsetup
```

6. **Record the line you created** in a safe place for later

### 3. Setup messaging API with Twilio

- **Why:**

Twilio is a cloud service that lets us send text messages programmatically. This is one of the few services we use that costs money, so you will have to sign up with a credit card and add some money to your account.

- **Steps:**

1. Create a twilio account: <https://www.twilio.com/try-twilio>

- We recommend setting up your account with *Auto Recharge*, so that your service will never go out because you forgot to check your Twilio balance.

2. From your *Console Dashboard* record the **Account SID** and **Auth Token** for later.

3. Purchase a number with *SMS* capabilities.

4. From the phone number configuration **record the *Phone Number*** field for later (make sure you don't copy the *Friendly Name*, we want a value that

#### Properties

FRIENDLY NAME	(971) 302-2591
SID	PNae79cfb138b4b2aba24d7018cabf722f
PHONE NUMBER	+19713022591
LOCATION	Portland, OR US
CAPABILITIES	Voice, Fax, SMS, MMS

looks like: +19713022591)

#### 4. Setup emailing API with Mailgun

- **Why:**

Mailgun is an automated mailing service that we will use to send on-boarding emails to users. It handles a lot of junk mail filtering and has very good free tier of up to 10,000 emails per month. It is very unlikely that you will exceed this with the app, but you will need to add a card to link a custom domain.

- **Steps:**

1. Create an account: <https://signup.mailgun.com/new/signup>
2. Got to the *Security* tab in your *Account Settings* and click the eyeball button next to your *Private API Key*. Then copy and **record your *Private API Key*** for later.
3. Add a new domain
  - Use which ever domain you would like have the server on, but prepend **mg.** to the beginning of the domain.

- *Example:*

If the domain you were using was:

twigbikeshare.com

The domain name that you would enter in mailgun would be:

4. After you add the domain, record the *Hostname* and *Value* for each of the

### 2. Add DNS Records For Sending

TXT records (known as SPF & DKIM) are **required to send and receive email** with Mailgun.

Type	Hostname	Enter This Value
TXT	mg.twigbikeshare.com	v=spf1 include:mailgun.org ~all
TXT	smtp._domainkey.mg.twigbikeshare.com	k=rsa; p=MIGfMA0GCSqGSIB3DQEBAQUAA4GNADCBiQKBgQCpMh1//bb24qWKVSSDjdXEiDK2s1fbjXvBRkl0IHxb6dPM4VVZ3kuu5G9yBKukVedTxiURHtYiBlvljoa+4W7buKhlM/gKT9GCzesjKHoojpb+O8B/tjMKddBkR3I5Ger/sYR4yUG771CHwT/2zGvbByxC/5PJDnr0ge45PRidQI

MX records are **recommended for all domains, even if you are only sending messages**. Unless you already have MX records for @mg.twigbikeshare.com pointing to another email provider (e.g. Gmail), you should update the following records. [More info on MX records](#)

Type	Priority	Enter This Value
MX	10	mx.a.mailgun.org
MX	10	mx.b.mailgun.org

two *TXT Records* for later.

## 5. Setup repository with Github

- **Why:**

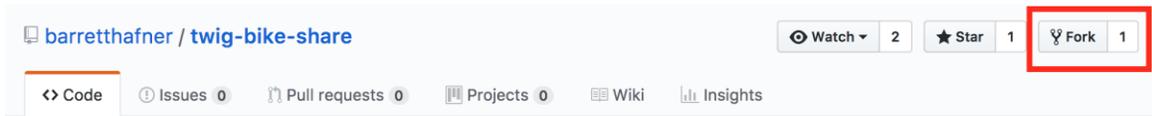
The code for this project lives on github.com at:

<https://github.com/barretthafner/twig-bike-share>. To install it you will need to “Fork” the repository to your own account and then connect it to Heroku (if you are an advanced user, you can just clone the repository and push it to Heroku manually).

- **Steps:**

1. Create a Github account: <https://github.com/join>
2. Verify your email address

3. Go to <https://github.com/barretthafner/twig-bike-share> and *Fork* the repo.



## 6. Setup application server with Heroku

- **Why:**

Heroku will run our app in the cloud. Taking care of uptime worries and streamlining the deployment process. This is one of the more complicated steps because we are hooking *most* of the other steps that we did into our app at this point.

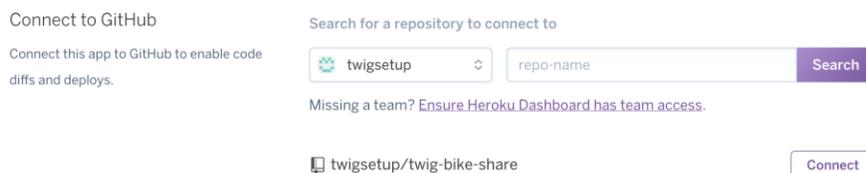
- **Steps:**

1. Create a Heroku account : <https://signup.heroku.com/>
2. Verify your email and create a password.
3. Create a new app
  - Choose whichever region you chose for your database.
4. Go to the *Settings* tab of your new app and click the *Reveal Config Vars* button.
5. Add the following config vars:

KEY	VALUE
APP_DOMAIN	The domain you plan to use for the app. <i>Ex:</i> <a href="https://twigbikeshare.com">twigbikeshare.com</a>
SESSION_SECRET	A long string of text that doesn't really matter, it's used to create unique identifiers. Just make up a sentence around 12 - 20 words.
DB_URL	The line of text you created when setting up Mlab in Step 2.
TWILIO_ACCOUNT_SID	The <i>Account SID</i> you recorded from Twilio in Step 3
TWILIO_AUTH_TOKEN	The <i>Auth Token</i> you recorded from Twilio in Step 3
TWILIO_NUMBER	The <i>Phone Number</i> you recorded from Twilio in Step 3
MAILGUN_PRIVATE_API_KEY	The <i>Private API Key</i> you recorded when setting up mailgun in Step 4.

KEY	VALUE
MAILGUN_DOMAIN	The domain you entered into mailgun in Step 4 <i>Ex:</i> mg.twigbikeshare.com
MAILGUN_EMAIL	This is the email address you want the app to send validation emails from. It can be anything so long as it ends in the domain of the app. <i>Ex:</i> <a href="mailto:noreply@twigbikeshare.com">noreply@twigbikeshare.com</a>
SITE_TITLE	The name that you would like to see on the home page and in the tab of the page in the browser.
SUPPORT_SITE	If your bike share is offering troubleshooting for users, you can add a URL here that will appear at the bottom of the home page.
SUPPORT_EMAIL	An email address that users can contact you at for support.
SUPPORT_TIME_ZONE	A time zone value for the app, select from this list: <a href="https://www.vmware.com/support/developer/vc-sdk/visdk41pubs/ApiReference/timezone.html">https://www.vmware.com/support/developer/vc-sdk/visdk41pubs/ApiReference/timezone.html</a>
SECURE	Just set this one to: true

6. Still in the *Settings* tab, scroll down to *Domains and certificates*, click the *Add a domain* button, and type in the domain that you are using. **Record the *DNS Target*** that appears for the next step.
7. In the *Deploy* tab, next to *Deployment method*, select Github, and then press the *Connect to GitHub* button.
8. Authorize Github, then still in the *Deploy* tab, next to *Connect to GitHub*, just leave the search bar blank and click the search button. The repo you forked



earlier should appear.

9. Press the *Connect* button. Then scroll down and press the *Deploy Branch* button.

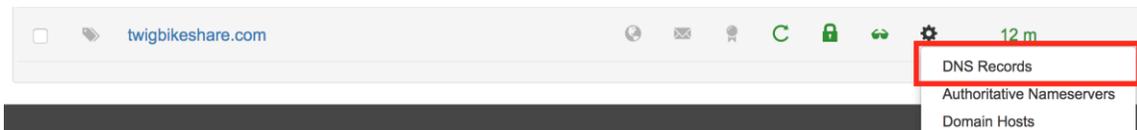
10. Wait ~1min for the dyno to deploy.
11. Go to the *Resources* tab and press the *Upgrade to Hobby...* button (you need to wait until the dyno is running so you might need to refresh the page a few times). Select the *Hobby* level and click the *Save* button (you may be asked for a credit card at this step).
12. If everything went as planned, you app should be running. If you click the *Open App* button at the top of the page it should open a new and you should be able to see the app Setup Page. **Don't get too excited!** We aren't quite done yet...

## 7. DNS configuration

- **Why**

To send emails with Mailgun we need to domain name to send emails as, so when a user gets an email they will see that it is from our service (Ex: [xxxxxx@twigbikeshare.com](mailto:xxxxxx@twigbikeshare.com)). It also will allow users to access our site directly from the purchased domain name. This guide will assume you are using Porkbun, but these steps should work for any domain name service. This is also the most error-prone and complicated part of the setup.

- **Steps**



1. Head over to your domain name provider and find the *DNS Records* setting.

Type	Host	Answer	TTL
CNAME	<i>leave blank</i>	the <i>DNS Target</i> you recorded in Step 6	300
TXT	the <i>Hostname</i> for the first <i>TXT Record</i> you recorded in Step 4	the <i>Value</i> of the first <i>TXT Record</i> you recorded in Step 4	300
TXT	the <i>Hostname</i> for the second <i>TXT Record</i> you recorded in Step 4	the <i>Value</i> of the second <i>TXT Record</i> you recorded in Step 4	300

2. Add the following DNS records:
3. It takes some time for these changes to propagate across the internet so give it about a minute.

4. You can check if these records took effect on Heroku by going to the *Settings* page of your app and under *Domains and certificates* clicking the *Refresh Status* button. A green checkmark should appear next to your domain and you should be able to navigate to your domain and see the app.
5. You can check if the records took effect for Mailgun by checking your *Domain Name Settings* under *Domain Verification & DNS* and clicking the *Check DNS Records Now* button. There should be checks next to the two *TXT Records*. The other records won't have checks next to them, but that won't matter for the purposes of this application.

## 8. Setup your app

- **Why**

Once all the above steps are completed we can initialize the app for use.

- **Steps**

1. When you load the app you should see the setup page, which will ask you for a *Name*, *Email address*, and *Password* make sure that you enter these carefully and record them. After you click submit, give the app a few seconds to initialize. You will automatically be redirected to the login page.
2. Once you see the login page, log in with the *Email* and *Password* you just entered.
3. The app is ready to use!

## IV. Contacts

- If you have any trouble setting up the bike share please reach out to the creator:  
Barrett Hafner // [barrett.git@gmail.com](mailto:barrett.git@gmail.com)

## V. Acknowledgements

- Thank you to the Westside Transportation Initiative (<http://www.wta-tma.org/>) for funding the evolution of this application and the creation of this setup guide.
- Thank you to all the past contributors of the Open Bike Initiative (<http://openbikeinitiative.org/>) this application was designed after a similar system built by OBI for a different bike pilot in 2013.